



PHÂN LỚP HÌNH ẢNH DỰA TRÊN CẤU TRÚC KD-TREE

Nguyễn Thị Định^{1,3}, Văn Thế Thành², Lê Mạnh Thanh^{1*}

¹ Khoa Công nghệ Thông tin, Trường ĐH Khoa học, Đại học Huế, 77 Nguyễn Huệ, Huế, Việt Nam

² Trường Đại học Sư phạm Tp. HCM, 280 An Dương Vương, Quận 5, Tp. Hồ Chí Minh, Việt Nam

³ Khoa Công nghệ Thông tin, Trường ĐH Công nghiệp Thực phẩm Tp. HCM, 140 Lê Trọng Tấn, Q. Tân Phú, Tp. Hồ Chí Minh, Việt Nam

Tóm tắt. Trong bài báo này, một phương pháp phân lớp hình ảnh dựa trên cấu trúc KD-Tree được đề xuất nhằm thực hiện phân lớp nhiều lần cho một ảnh đầu vào theo mô hình phân lớp đa tầng. Chúng tôi đã đề xuất các thuật toán phân lớp hình ảnh dựa trên cấu trúc KD-Tree, huấn luyện bộ véc-tơ phân lớp và mô hình phân lớp ảnh. Dựa trên cơ sở lý thuyết này, chúng tôi thực nghiệm trên các bộ ảnh COREL, Wang, CIFAR-100, Caltech101, Clatech256 với hai nhóm đặc trưng SCH, SCH36 và so sánh với các công trình khác cùng bộ dữ liệu để chứng minh tính khả thi của phương pháp. Kết quả thực nghiệm cho thấy phương pháp của chúng tôi là hiệu quả và có thể áp dụng được cho các hệ phân loại hình ảnh thuộc các lĩnh vực khác nhau. Đồng thời, đây là một hướng tiếp cận mới trên cấu trúc KD-Tree áp dụng cho bài toán phân lớp hình ảnh được đánh giá với hiệu suất cao hơn các phương pháp khác thực hiện trên cùng bộ ảnh.

Từ khóa: phân lớp ảnh, mô hình phân lớp, hiệu suất phân lớp, KD-Tree

Image classification based on KD-Tree structure

Nguyen Thi Dinh^{1,3}, Van The Thanh², Le Manh Thanh^{1*}

¹ Faculty of Information Technology, University of Sciences, Hue University,
77 Nguyen Hue St., Hue, Vietnam

² HCMC University of Education, 280 An Duong Vuong St., District 5, Ho Chi Minh City, Vietnam

³ Faculty of Information Technology, Ho Chi Minh City University of Food Industry,
140 Le Trong Tan St., Tan Phu District, Ho Chi Minh City, Vietnam

Abstract. In this paper, a method of image classification based on a KD-Tree structure is proposed to perform multiple classifications for an input image according to the multi-layer classification model. The process of image classification based on the KD-Tree structure is conducted with the method of building a KD-Tree structure and training a set of classifier vectors. Therefore, some algorithms, including an image classification algorithm based on the KD-Tree structure, classifier vector training algorithm, and image classification models, are proposed. Based on this theory, the experiment is built on the image sets COREL, Wang,

* Liên hệ: lmthanh@hueuni.edu.vn

CIFAR-100, Caltech101, and Clatech256 with two groups of features: SCH and SCH36 and compared with other works with the same data set to demonstrate the feasibility of the proposed method. The results show that our method is effective and can be applied to image classification systems in different fields. At the same time, this is a new approach to the KD-Tree structure, applied to the image classification problem with higher performance than other methods with the same set of images.

Keywords: image classification, classification model, classification performance, KD-Tree

1 Giới thiệu

Phân lớp hình ảnh là một chủ đề quan trọng trong lĩnh vực thị giác máy tính và thu hút sự quan tâm của các nhóm nghiên cứu trong thời gian qua. Một mô hình phân lớp có thể được xây dựng bằng các thuật toán: *rừng ngẫu nhiên*, *cây quyết định*, *mạng học sâu* và nhiều kỹ thuật khác. Bài toán phân loại đối tượng bằng hình ảnh được ứng dụng trong nhiều lĩnh vực như nhận dạng mẫu, phân loại bệnh nhân, dự đoán thị trường chứng khoán và ứng dụng trong nhiều lĩnh vực khác. Ngày nay, sự đa dạng của ảnh số làm cho quá trình nhận dạng đối tượng qua hình ảnh gặp nhiều khó khăn, đặc biệt là các bộ ảnh đa đối tượng [1, 2]. Vì vậy, nâng cao hiệu suất cho mô hình phân loại đối tượng bằng hình ảnh là cần thiết. Trong bài báo này, chúng tôi đánh giá mô hình phân loại đối tượng bằng hình ảnh qua các giai đoạn: xây dựng, huấn luyện mô hình, kiểm thử và chọn tham số để kết quả phân loại từ mô hình đạt hiệu quả cao.

Mô hình phân lớp ảnh được xây dựng dựa trên một tập dữ liệu gán nhãn gọi là tập huấn luyện. Nhiệm vụ của bài toán phân lớp là cần tìm một mô hình phân lớp để khi có dữ liệu mới thì có thể xác định được dữ liệu đó thuộc vào phân lớp nào [3, 18, 19]. Hiện nay, có nhiều bài toán phân lớp dữ liệu, như phân lớp nhị phân (binary classification) và phân lớp đa lớp (multiclass classification), được thực hiện với nhiều kỹ thuật khác nhau như k -NN (*k-Nearest Neighbors*) [5, 14, 20], SVM (*Support Vector Machines*) [15], CNN (*Convolutional Neural Networks*) [15] và DNN (*Deep Neural Networks*) [16]. Trong bài báo này, một phương pháp phân lớp đối tượng bằng hình ảnh dựa trên cấu trúc KD-Tree được xây dựng để hình thành các phân lớp tại nút lá. Quá trình phân lớp này được thực hiện tại mỗi tầng trên cây KD-Tree để hình thành tập véc-tơ phân lớp từ nút gốc đến nút trong thuộc tầng sát nút lá. Tập véc-tơ phân lớp này được điều chỉnh thông qua quá trình huấn luyện trọng số để tìm ra tập giá trị tốt nhất cho mô hình phân lớp dựa trên KD-Tree.

Đóng góp bài báo gồm: (1) Mô tả quá trình phân lớp ảnh bằng cấu trúc KD-Tree; (2) Xây dựng thuật toán phân lớp hình ảnh dựa trên cấu trúc KD-Tree; (3) Huấn luyện bộ véc-tơ phân lớp; (4) Đề xuất mô hình phân lớp ảnh số dựa trên cấu trúc KD-Tree; (5) Đánh giá mô hình bằng

thực nghiệm trên các bộ ảnh COREL [8], Wang [9], CIFAR-100 [10], Caltech101 [11] và Clatech256 [12].

2 Các công trình liên quan

Ngày nay, phân lớp hình ảnh đã được nhiều tác giả thực hiện với các kỹ thuật khác nhau như phân lớp bằng thuật toán tìm kiếm láng giềng gần nhất (k-NN), mạng nơron tích chập (CNN), mạng học sâu (DNN), thuật toán SVM, Naïve Bayes, phân lớp theo cấu trúc cây quyết định (Decision Tree) và cây tìm kiếm đa chiều (KD-Tree). Trong phần này, chúng tôi tiến hành khảo sát một số phương pháp phân lớp hình ảnh để rút ra ưu, nhược điểm và so sánh với phương pháp phân lớp ảnh bằng KD-Tree, cụ thể là:

Khotimah và cs. [22] đã thực hiện một phương pháp phân lớp hình ảnh bằng thuật toán cây quyết định trên mẫu cá ngừ đại dương với đặc trưng hình ảnh được trích xuất gồm kết cấu, hình dạng tỷ lệ giữa diện tích đầu và diện tích hình tròn, tỷ lệ hình tròn của đầu cá ngừ. Kết quả thực nghiệm đánh giá trên 60 mẫu ảnh cá ngừ loại mắt to, vây vàng và cá vằn với độ chính xác phân lớp là 0,88. Bên cạnh đó, Gautam và cs. [24] thực hiện một phương pháp phân lớp bằng thuật toán Naïve Bayes để phân loại tự động bạch cầu bằng cách sử dụng đặc điểm hình thái đối tượng thông qua hình ảnh. Kết quả thực nghiệm với độ chính xác phân loại cho 68 mẫu hình ảnh là 0,8088 và thời gian phân lớp cho mỗi đối tượng bằng hình ảnh trung bình là 22 giây. Đây là hai công trình thực hiện phân lớp hình ảnh bằng kỹ thuật cổ điển sử dụng Decision Tree và thuật toán Naïve Bayes đã mang lại kết quả khả quan. Tuy nhiên, số mẫu thực nghiệm để đánh giá hiệu suất phân lớp trên hai công trình này chưa lớn và chưa áp dụng cho phân loại hình ảnh đối với các tập ảnh thuộc nhiều lĩnh vực khác nhau. Vì vậy, trong bài báo này, chúng tôi tiến hành thực nghiệm phân loại hình ảnh bằng KD-Tree được thực nghiệm trên các bộ dữ liệu có tính chất khác nhau, tập ảnh thực nghiệm lớn để đánh giá chất lượng của phương pháp đề xuất.

Một hướng tiếp cận phân lớp hình ảnh bằng cách kết hợp nhiều kỹ thuật học máy như k-NN, CNN hoặc kết hợp kỹ thuật học máy với một cấu trúc dữ liệu và các nhóm đặc trưng hình ảnh trên các bộ ảnh COREL, Wang, CIFAR-100, Caltech101 đã được chúng tôi tiến hành thông qua một số công trình tiêu biểu. Nhi cs. [5] đã thực hiện một phương pháp phân lớp hình ảnh bằng cách kết hợp cấu trúc cây C-Tree và thuật toán phân lớp k-NN tại nút lá trên C-Tree đã thu được kết quả khả quan. Trong công trình này, các tác giả đã thực nghiệm trên bộ ảnh COREL và Wang với hiệu suất phân lớp là 0,7302 và 0,6894. Sự kết hợp một cấu trúc lưu trữ C-Tree và thuật toán phân lớp k-NN đã nâng cao hiệu suất cho quá trình phân lớp hình ảnh so với thuật toán k-NN. Đây là cơ sở để chúng tôi tiếp cận phương pháp phân lớp bằng KD-Tree để so sánh hiệu suất trên cùng bộ ảnh. Bên cạnh đó, Kumar và cs. [15] kết hợp thuật toán phân lớp SVM và k-NN trên các nhóm đặc trưng để tìm ra kết quả phân lớp tốt nhất trên tập ảnh Wang. Trong công trình

này, các tác giả đã thực hiện quá trình phân lớp ảnh bằng phương pháp SVM-NPP-SURF với hiệu suất phân lớp trên tập ảnh Wang là 0,79. Ngoài ra, các tác giả còn thực nghiệm trên các tập ảnh khác và kết hợp nhiều phương pháp nhằm đánh giá và so sánh với các công trình cùng lĩnh vực. Trong công trình này, ngoài kỹ thuật phân lớp được lựa chọn, các tác giả còn quan tâm đến vấn đề trích xuất đặc trưng hình ảnh phù hợp với từng tập dữ liệu cụ thể. Dựa trên ý tưởng lựa chọn đặc trưng hình ảnh từ công trình này, chúng tôi đã thực hiện một phương pháp phân lớp ảnh bằng KD-Tree kết hợp với nhóm các đặc trưng hình ảnh SCH và SCH36 để so sánh hiệu suất trên cùng bộ ảnh Wang. Kumar và cs. [11] đã thực hiện quá trình phân lớp hình ảnh bằng mạng CNN và thực nghiệm trên bộ dữ liệu CIFAR-100 trên 10 bộ dữ liệu mẫu gồm Bed, Bicycle, Bus, Chair, Couch, Motorcycle, Street Car, Table, Train và Wardrobe với độ chính xác phân lớp là 0,5802. Phương pháp được đánh giá là khả thi, nhưng các tác giả chưa công bố kết quả thực nghiệm trên 100 phân lớp của bộ ảnh này. Mặc khác, Dang và cs. [12] đã thực hiện bài toán phân lớp hình ảnh bằng thuật toán QKNN (Quantum k-Nearest Neighbor) – quá trình kết hợp thuật toán phân lớp k-NN với quá trình trích xuất véc-tơ đặc trưng hình ảnh. Sự kết hợp này được thực nghiệm trên bộ ảnh Caltech101 với 9 phân lớp có nhiều hơn 100 mẫu ảnh để thực nghiệm; tổng số ảnh thực nghiệm là 2921 với hiệu suất phân lớp 0,78 khi $k = 3$. Trong công trình này, các tác giả chưa công bố kết quả thực nghiệm trên toàn bộ tập ảnh để có cơ sở phân tích ưu nhược điểm của phương pháp.

Để minh họa cho phương pháp phân lớp bằng cấu trúc KD-Tree kết hợp với thuật toán k-NN, Zhang và cs. [21] sử dụng phương pháp phân lớp ảnh số để nhận diện khuôn mặt dựa trên phương pháp phân lớp hình ảnh kết hợp thuật toán k-NN và cấu trúc KD-Tree. Trong công trình này, mỗi hình ảnh trong tập dữ liệu tại pha huấn luyện được chia thành nhiều vùng; cấu trúc KD-Tree được xây dựng dựa trên tập ảnh phân vùng nhằm phân lớp dữ liệu hình ảnh. Tại pha kiểm thử, mỗi ảnh được chia thành nhiều phân vùng và cấu trúc KD-Tree được sử dụng trong quá trình tìm kiếm theo thuật toán k-NN. Trong công trình này, thuật toán KD-Tree được xây dựng theo cấu trúc chỉ mục đa chiều nhằm giảm thời gian tìm kiếm. Thực nghiệm trên bộ ảnh CUFS (Chinese University Face Sketch) chứng minh rằng phương pháp đề xuất của nhóm tác giả là hiệu quả. Đây là công trình mà trong đó các tác giả sử dụng cấu trúc KD-Tree kết hợp thêm kỹ thuật học máy để phân loại hình ảnh, nhưng hiệu suất đạt được chưa cao. Điều này là động lực cho phương pháp tiếp cận phân lớp hình ảnh trên cấu trúc KD-Tree được chúng tôi thực hiện trên cơ sở kế thừa cấu trúc KD-Tree nguyên thủy để cải tiến thành cấu trúc KD-Tree đa nhánh cân bằng theo hướng phân lớp hình ảnh và kế thừa từ công trình [6, 7].

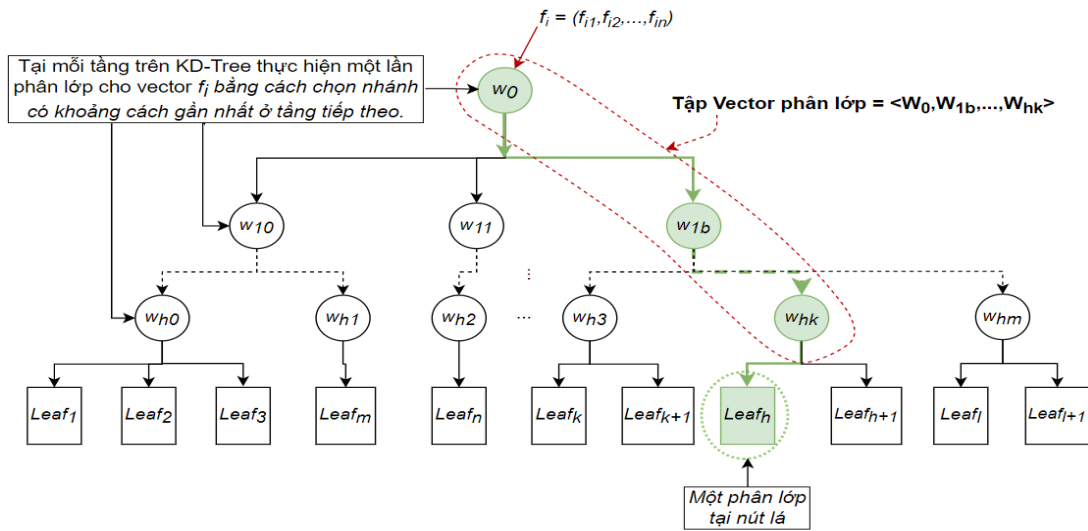
Các công trình nghiên cứu liên quan về các kỹ thuật phân lớp ảnh số được thực hiện theo nhiều phương pháp với các bộ ảnh thực nghiệm COREL, Wang, CIFAR-100 và Caltech101 và đã mang lại hiệu suất khác nhau. Tuy nhiên, các phương pháp này còn những hạn chế về thời gian huấn luyện và kết quả thực nghiệm chưa được công bố đầy đủ. Vì vậy, trong bài báo này, chúng

tôi đề xuất phương pháp phân lớp hình ảnh sử dụng cấu trúc KD-Tree trên năm bộ ảnh khác nhau về tính chất, số lượng phân lớp và số lượng hình ảnh nhằm phản ánh các ưu, nhược điểm để làm cơ sở so sánh với các công trình trên. Đồng thời, đây là một phương pháp được đánh giá là khả thi và có tính mới.

3 Phương pháp phân lớp hình ảnh dựa trên cấu trúc KD-Tree

3.1 Cấu trúc KD-Tree phân lớp

KD-Tree phân lớp được xây dựng là một cây đa nhánh cân bằng, dựa trên tính chất cây KD-Tree nguyên thủy [4], gồm một nút gốc lưu trữ véc-tơ phân lớp (w_0); tập nút trong $\{Node\}$, mỗi nút trong lưu trữ một véc-tơ phân lớp (w_{ik}) và tập nút lá $\{Leaf\}$, mỗi nút lá chứa nhiều véc-tơ hình ảnh $\{F = \langle f_1, f_2, \dots, f_n \rangle\}$, được mô tả và kế thừa từ công trình [6, 7]. Trong cấu trúc KD-Tree phân lớp xây dựng theo thuật toán CKDT [7], mỗi nút lá chứa tập véc-tơ hình ảnh thuộc nhiều phân lớp khác nhau. Vì vậy, mỗi nút lá được gán một nhãn theo số lượng véc-tơ cùng nhãn lớp là nhiều nhất theo thuật toán gán nhãn SL2L [7]. Cấu trúc KD-Tree và quá trình phân lớp trên KD-Tree được minh họa trên Hình 1.



Hình 1. Quá trình phân lớp trên cấu trúc KD-Tree cho véc-tơ f_i

3.2 Phương pháp phân lớp ảnh dựa trên cấu trúc KD-Tree

Quá trình phân lớp ảnh trên KD-Tree được thực hiện dựa vào Hình 1 theo quy tắc: Với một ảnh đầu vào (I) có véc-tơ đặc trưng $f_i = (f_{i1}, f_{i2}, \dots, f_{im})$ lần lượt đi qua nút gốc, nút gốc lưu trữ một véc-tơ trọng số (w_0). Tại nút gốc, giá trị đầu ra được tính theo hàm mục tiêu Sigmoid là $y_{i0} =$

Sigmoid($f_i \times w_i$) [7]; giá trị đầu ra của véc-tơ f_i tại mỗi tầng trên KD-Tree là cơ sở cho quá trình chọn nhánh với khoảng cách gần nhất ở tầng tiếp theo. Khoảng cách gần nhất được xác định từ vị trí y_{ij} đến vị trí w_{nk} . Quá trình này lặp lại cho các tầng tiếp theo đến khi gặp nút lá $Leaf_i$ và nhãn của nút lá này chính là nhãn lớp của ảnh đầu vào được phân lớp trên KD-Tree. Phương pháp phân lớp ảnh dựa trên cấu trúc KD-Tree được thực hiện nhiều lần cho một ảnh đầu vào do cây KD-Tree được xây dựng là cây đa tầng. Tại mỗi tầng trên KD-Tree thực hiện một lần phân lớp nên hiệu suất phân lớp ảnh trên KD-Tree là khá cao.

Hàm mục tiêu: Trong quá trình huấn luyện mô hình phân lớp dựa trên cấu trúc KD-Tree, cần xác định giá trị đầu ra tại mỗi nút *trong* của cây. Do đó, hàm mục tiêu Sigmoid (công thức (1)) được sử dụng là hàm tuyến tính tăng trên miền [0..1]. Đồng thời, hàm Sigmoid luôn có đạo hàm tại mọi điểm trên miền [0..1] nhằm tăng hiệu quả trong quá trình huấn luyện véc-tơ trọng số khi lấy đạo hàm theo công thức (2) và tiết kiệm chi phí huấn luyện cấu trúc KD-Tree phân lớp .

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{1}$$

$$\sigma'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1}{1+e^{-x}} \frac{e^{-x}}{1+e^{-x}} = \sigma(x)(1-\sigma(x)) \tag{2}$$

Gọi $w_i = (w_{i1}, w_{i2}, \dots, w_{in})$ là véc-tơ trọng số lưu trữ tại $Node_i$, công thức (3) cập nhật giá trị véc-tơ trọng số tại mỗi $Node_i$ theo hàm $\sigma(W_i * f_i)$ và η là hằng số cho trước như sau:

$$W_i = W_i - \sigma(W_i * f_i) * \eta * \frac{\partial(\sigma(W_i * f_i))}{\partial(W_i)} = W_i - \sigma(W_i * f_i) * \eta * \sigma(W_i * f_i) * (1 - \sigma(W_i * f_i)) \tag{3}$$

Sau khi thực hiện gán nhãn cho tất cả các nút lá, tính hiệu suất phân lớp trên KD-Tree với tập véc-tơ trọng số khởi tạo theo công thức (4).

$$P_m = Avg\left(\frac{\sum(f_{ki})}{\sum(f_{mi})}\right) \tag{4}$$

trong đó $\sum(f_{ki})$ là tổng số véc-tơ đúng nhãn lớp thuộc nút lá $Leaf_i$; $\sum(f_{mi})$ là tổng số véc-tơ thuộc nút lá $Leaf_i$.

Sau mỗi lần cập nhật véc-tơ phân lớp theo công thức (3), hiệu suất phân lớp trên KD-Tree được tính theo công thức (4). So sánh nếu $P_j > P_i$ ($j > i$) thì chọn phương án cập nhật véc-tơ phân lớp mới; trong trường hợp ngược lại, giữ nguyên véc-tơ phân lớp trước đó.

Để quá trình huấn luyện tập véc-tơ phân lớp trên KD-Tree giảm chi phí về mặt thời gian cũng như các chi phí xây dựng cây KD-Tree, cần thực hiện các quy tắc sau:

Quy tắc 1: Gọi $F = \{f_i : f_i \in Node_i.left; i = 1..k\}$ là tập các vector thuộc nhánh trái của $Node_i$. Muốn di chuyển một vector $f_i \in F$ sang nhánh phải của $Node_i$ ($Node_i.right$) thì chọn vector f_j thỏa mãn điều kiện: $|Sigmoid(Node_i.w_i * f_i)| \leq |Sigmoid(Node_i.w_i * f_j)|, \forall f_j \in F$

Quy tắc 2: Gọi $F = \{f_i : f_i \in Node_i.right; i = 1..k\}$ là tập các vector thuộc nhánh phải của $Node_i$. Muốn di chuyển một vector $f_i \in F$ sang nhánh trái của $Node_i$ ($Node_i.left$) thì chọn vector f_j thỏa mãn điều kiện: $|Sigmoid(Node_i.w_i * f_i)| \leq |Sigmoid(Node_i.w_i * f_j)|, \forall f_j \in F$

Quy tắc 3: Gọi $L = \{f_i : f_i \in leaf_k; i = 1..m\}$ là tập các vector sai đường đi tại nút lá $leaf_k$. Khi đó, ưu tiên chọn vector f_i có vị trí sai đường đi tại $Node_i$ để điều chỉnh trước thỏa mãn điều kiện: $Node_i.level > Node_k.level; \forall Node_i, Node_k \in f_i.Path$

Quy tắc 4: Gọi $Node_i$ là vị trí cần điều chỉnh vector trọng số $Node_i.w_i$. Khi thay đổi giá trị vector $Node_i.w_i$ thì chỉ cần tạo lại phần cấu trúc KD-Tree tính từ vị trí $Node_i$ đến tầng nút lá là con của $Node_i$. Điều này làm giảm đi chi phí xây dựng KD-Tree đáng kể so với phải xây dựng lại KD-Tree từ nguyên tập dữ liệu ban đầu.

Thuật toán 1. Thuật toán phân lớp ảnh trên cấu trúc KD-Tree (CLKDT)

Input: Véc-tơ đặc trưng f_i của ảnh I,

Output: Tên phân lớp CL_I của ảnh I;

Function CLKDT (f_i , KD-Tree)

Begin

For (int h = 0; h < chiều cao KD-Tree - 1; h++) **do**

y_{ij} = Tính giá trị đầu ra của f_i tại $Node_j$;

So sánh giá trị y_{ij} với giá trị tại các $Node_k$ ở tầng (h+1);

f_i đi theo nhánh con có khoảng cách ngắn nhất từ y_{ij} đến $Node_k$;

EndFor;

If ($f_i \in Leaf_k$) **then** $f_i.label = Leaf_k.label; CL_I = Leaf_k.label$;

Return CL_I ;

End.

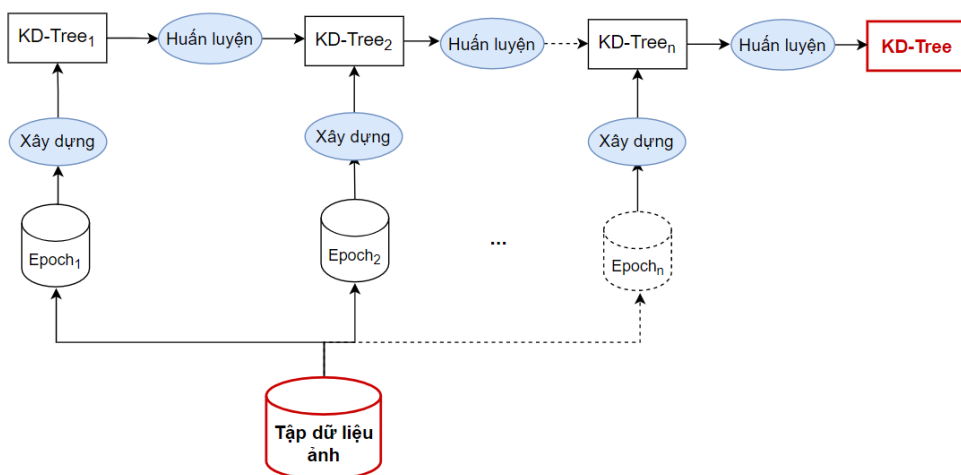
Gọi h là chiều cao của cấu trúc KD-Tree; k là số nhánh tối đa tại $Node_i$ bất kỳ thì dữ liệu đầu vào là vector đặc trưng f_i có n chiều. Khi truyền vector f_i vào KD-Tree, thuật toán 1 duyệt qua các mức của cây. Tại mỗi mức trên KD-Tree, chọn một nút tốt nhất và đi theo hướng đã chọn. Do đó,

tại mỗi mức có tối đa k phép so sánh để chọn nút tốt nhất. Trong mỗi lần so sánh, thuật toán 1 duyệt qua n phần tử của vector f_i . Vì vậy, tại mỗi mức, số phép toán tối đa là $k \times n$. Cây có chiều cao h , nên số phép toán tối đa để duyệt từ gốc đến lá theo một hướng được chọn là $k \times n \times h$. Vì h và k là hằng số nhỏ, nên độ phức tạp của thuật toán phụ thuộc vào n . Mặc khác, số chiều vector f_i là cố định ban đầu nên n cũng là hằng số. Gọi C là giá trị hằng số và $C > k \times h \times n$ nên $k \times h \times n \leq C \times 1$. Vì vậy, độ phức tạp của thuật toán 1 là $O(1)$.

3.3 Huấn luyện tập véc-tơ phân lớp trên cấu trúc KD-Tree

Sau khi xây dựng cấu trúc KD-Tree phân lớp, mỗi nút lá là một phân lớp hình ảnh. Ban đầu, tập véc-tơ phân lớp cho mỗi nút lá được khởi tạo ngẫu nhiên nên hiệu suất phân lớp trên KD-Tree chưa cao. Do đó, cần phải thực hiện quá trình điều chỉnh các véc-tơ phân lớp lưu trữ tại các *Node*. Quá trình điều chỉnh véc-tơ phân lớp này được gọi là quá trình huấn luyện tập véc-tơ phân lớp trên cấu trúc KD-Tree.

Quá trình huấn luyện để tìm ra tập véc-tơ phân lớp tốt nhất cho mô hình phân lớp dựa vào cấu trúc KD-Tree được thực hiện theo sơ đồ trên Hình 2. Để giảm chi phí cho quá trình xây dựng KD-Tree và huấn luyện tập véc-tơ phân lớp, tập dữ liệu ảnh ban đầu gồm n phần tử được chia thành nhiều *Epoch* dữ liệu ngẫu nhiên. Mỗi *Epoch_i*, chứa m phần tử ($m < n$), thực hiện xây dựng cây KD-Tree và huấn luyện tìm tập véc-tơ phân lớp tốt nhất F_{max_i} . Sau đó, từ tập F_{max_i} , tiếp tục xây dựng cây KD-Tree cho *Epoch_(i+1)* và huấn luyện tìm tập $F_{max(i+1)}$. Quá trình này lặp lại cho đến khi hết các *Epoch* dữ liệu và quá trình xây dựng cây KD-Tree và huấn luyện tìm tập véc-tơ phân lớp dừng lại. Kết quả của quá trình này là cấu trúc KD-Tree cho kết quả phân lớp tốt nhất ứng với bộ



Hình 2. Sơ đồ xây dựng và huấn luyện KD-Tree theo các Epoch_i dữ liệu

véc-tơ phân lớp tốt nhất (F_{\max}) trên từng bộ ảnh thực nghiệm. Quá trình huấn luyện bằng cách chia nhỏ bộ dữ liệu giúp giảm chi phí thời gian huấn luyện so với việc huấn luyện từ nguyên bộ dữ liệu ban đầu.

Mô hình phân lớp dựa trên cấu trúc KD-Tree được huấn luyện theo từng bộ dữ liệu ($Epoch_i$) diễn ra song song với quá trình tạo cây. Véc-tơ phân lớp $w_i = (x_{i1}, x_{i2}, \dots, x_{in})$ tại $Node_i$ được huấn luyện theo đạo hàm ngược hướng Gradient nhằm giảm sai số trung bình dựa trên hàm truyền Sigmoid. Thuật toán huấn luyện véc-tơ phân lớp trên KD-Tree được trình bày như sau:

Thuật toán 2. Thuật toán huấn luyện tập véc-tơ phân lớp trên KD-Tree (TVKDT)

Input: Bộ Véc-tơ khởi tạo InitWeight

Output: Bộ Véc-tơ huấn luyện;

Function TVKDT (InitWeight, Epoch_i)

Begin

Vector = InitVector;

Repeat

- (1) Tạo KD-Tree theo Epoch_i với bộ Vector khởi tạo, chiều cao h, n nhánh;
- (2) Gán nhãn cho các nút lá trên KD-Tree;
- (3) P_i = Tính hiệu suất phân lớp trên KD-Tree với bộ Vector khởi tạo;
- (4) Xác định đường đi sai của những vector f_k trên KD-Tree;
- (5) Xác định đường đi đúng của những vector f_k bị sai đường đi;
- (6) Tìm vị trí $Node_i$ làm sai đường đi của vector f_k ;
- (7) TrainVector = Điều chỉnh vector lưu tại $Node_i$ để f_k đúng đường đi;
- (8) Tạo lại KD-Tree theo bộ vector đã điều chỉnh;
- (9) Gán nhãn cho các nút lá trên KD-Tree vừa mới tạo;
- (10) P_j = Tính hiệu suất phân lớp sau khi điều chỉnh vector tại $Node_i$;

Until ($P_j < P_i$);

Vector = TrainVector;

Return Vector;

End.

Thuật toán 2 được thực hiện lần lượt các bước (1) đến (10) gồm:

- (1) Tạo KD-Tree độ phức tạp $O(N)$, trong đó N là số phần tử tham gia tạo KD-Tree
- (2) Số nút lá k^h cần gán nhãn, độ phức tạp là $O(k^h)$
- (3) Duyệt trên từng vector, độ phức tạp là $O(N)$
- (4) Trong trường hợp xấu nhất, tất cả vector đều sai so với gán nhãn ban đầu, phải duyệt qua tất cả các vector từ nút lá trở về gốc (chiều cao h) nên số phép toán là $h \times N$
- (5) Trong trường hợp xấu nhất, phải duyệt qua tất cả các vector từ nút lá trở về gốc nên số phép toán là $h \times N$
- (6) Trong trường hợp xấu nhất, vị trí $Node_i$ sai là nút gốc, nên số phép toán là $h \times N$
- (7) Trong trường hợp xấu nhất, cần điều chỉnh hết các vector trọng số thì số phép toán là: $T = (k^0 + k^1 + \dots + k^n) * n = \left(\frac{k^{h+1} - 1}{k - 1}\right) * n = \left(\frac{n}{k - 1}\right) * (k^{h+1} - 1) \approx O(k^h)$
- (8) Lặp lại bước (1), độ phức tạp $O(N)$
- (9) Lặp lại bước (2), độ phức tạp là $O(k^h)$
- (10) Lặp lại bước (3), độ phức tạp là $O(N)$

Số phép toán sau một quá trình là:

$$(N + k^h + N + h * N + h * N + h * N + T + N + k^h + N) = (3h + 4) * N + 2k^h + T \approx O(k^h)$$

Vì vậy, độ phức tạp của thuật toán 2 là: $O(k^h)$

Trong thực nghiệm, bộ ảnh Caltech256 có số phân lớp nhiều nhất là 257 nên chọn $k = 5$ và $h = 4$ tức là có thể tạo ra 625 phân lớp; điều này là khả thi.

Sau khi huấn luyện, tập véc-tơ phân lớp trên KD-Tree cho mỗi bộ ảnh thực nghiệm là khác nhau. Hình 3 minh họa tập véc-tơ phân lớp cho bộ ảnh COREL sau khi đã huấn luyện để hiệu suất phân lớp đạt 0,8672 trên KD-Tree với chiều cao bằng 2 và số nhánh bằng 4.

Row	Col	Value
1	0	(0.00466, 0.0070, 0.00658, 0.00862, 0.00592, 0.00304,
2	1	(0.00581, 0.0094, 0.00624, 0.00878, 0.00794, 0.00404,
3	2	(0.00708, 0.0075, 0.00982, 0.00805, 0.00718, 0.010780341,
4	3	(0.00854, 0.00832, 0.00298, 0.00209, 0.00942, 0.00697,
5	4	(0.00404, 0.00975, 0.00675, 0.00808, 0.00844, 0.0099, 0.00132,

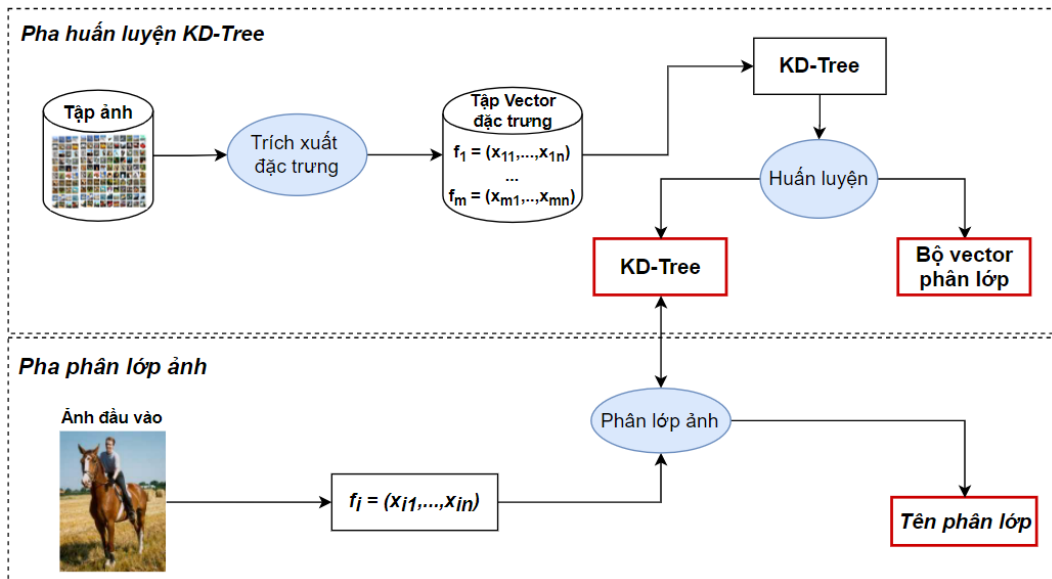
Hình 3. Minh họa tập véc-tơ phân lớp trên KD-Tree cho bộ ảnh COREL

3.4 Mô hình phân lớp ảnh dựa trên cấu trúc KD-Tree

Mô hình phân lớp ảnh dựa trên cấu trúc KD-Tree được thực hiện theo hai pha: Pha huấn luyện KD-Tree và pha phân lớp hình ảnh gồm các bước:

- (1) Trích xuất véc-tơ đặc trưng cho tập dữ liệu Training và huấn luyện cấu trúc KD-Tree.
- (2) Thực hiện huấn luyện bộ véc-tơ phân lớp sau khi xây dựng cấu trúc KD-Tree theo các *Epoch*. Kết quả của quá trình huấn luyện là thu được cấu trúc KD-Tree và bộ véc-tơ phân lớp để cho kết quả phân lớp ảnh trên tập dữ liệu ảnh ban đầu là tốt nhất.
- (3) Trích xuất véc-tơ đặc trưng của ảnh đầu vào cần thực hiện phân lớp.

Thực hiện phân lớp cho ảnh đầu vào dựa trên cấu trúc KD-Tree đã huấn luyện. Kết quả của quá trình phân lớp này là tên phân lớp cho ảnh đầu vào.



Hình 4. Mô hình phân lớp ảnh dựa trên cấu trúc KD-Tree

4 Thực nghiệm và đánh giá kết quả thực nghiệm

4.1 Dữ liệu thực nghiệm

Để đánh giá hiệu năng của các hệ thống phân lớp hình ảnh, chúng tôi sử dụng một số bộ dữ liệu ảnh tiêu chuẩn cho quá trình thực nghiệm gồm COREL, Wang, CIFAR-100, Caltech101 và Caltech256. Tập ảnh thực nghiệm được chia thành hai phần: 70% số lượng ảnh dùng để huấn luyện mô hình phân lớp (Training) và 30% số lượng ảnh dùng cho kiểm thử mô hình (Testing) (Bảng 1). Các bộ dữ liệu này có số lượng ảnh và số phân lớp chênh lệch nhau khá lớn; khác nhau

về tính chất nhằm đánh giá hết ưu nhược điểm của phương pháp đề xuất. Mỗi bộ ảnh có số phân lớp và số lượng ảnh khác nhau ảnh hưởng đến thời gian huấn luyện cấu trúc KD-Tree.

Bảng 1. Mô tả các tập dữ liệu ảnh thực nghiệm

Bộ ảnh	Số lượng ảnh	Số ảnh Training	Số ảnh Testing	Số phân lớp
COREL	1.000	700	300	10
WANG	10.800	7.560	3.240	80
CIFAR-100	60.000	42.000	18.000	100
Caltech101	9.144	6.500	2.644	102
Caltech256	30.607	21.424	9.183	257

Quá trình trích xuất véc-tơ đặc trưng hình ảnh được thực hiện theo thuật toán 3 như sau:

Thuật toán 3. Trích xuất đặc trưng hình ảnh (EFI)

Input: Ảnh đầu vào I

Output: Véc-tơ đặc trưng f_i của ảnh I

Function EFI (I)

Begin

$$f_i = \phi;$$

Foreach (image I) **do**

 Phân đoạn ảnh I;

Foreach (phân đoạn I_i của ảnh I) **do**

 m1 = Trích xuất các giá trị màu MPEG7;

 m2 = Trích xuất các giá trị đặc trưng SIFT MPEG7;

 m3 = Trích xuất các giá trị đặc trưng SHI MPEG7;

 m4 = Trích xuất các giá trị màu Color Newton;

 m5 = Trích xuất các giá trị đặc trưng HOG;

 m6 = Trích xuất các giá trị đặc trưng Sobel HOG;

 m7 = Trích xuất các giá trị đặc trưng Laplace HOG;

EndForeach;

EndForeach;

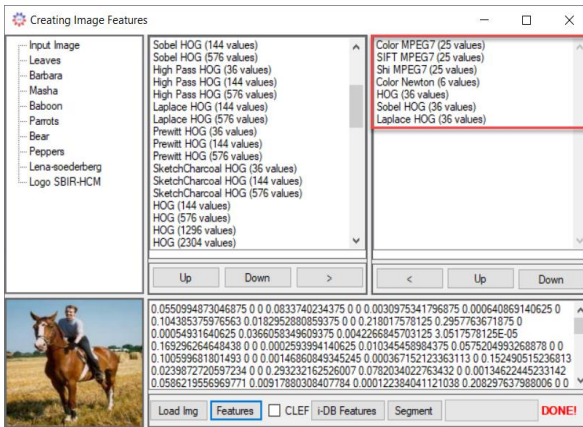
$$f_i = m_1 \cup m_2 \cup m_3 \cup m_4 \cup m_5 \cup m_6 \cup m_7;$$

Return f_i ;

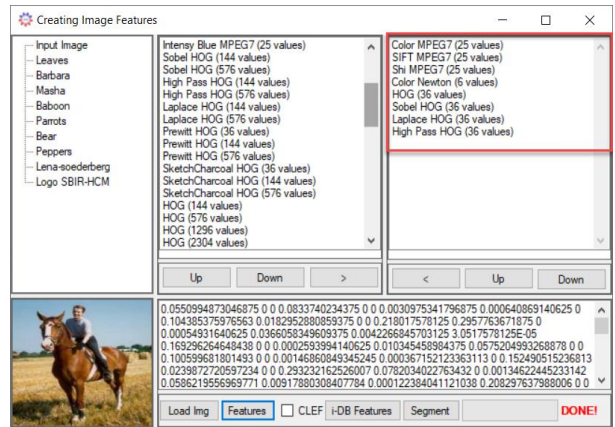
End.

Độ phức tạp của thuật toán EFI là $O(n)$. Trong đó n là số phân đoạn của ảnh I vì n là hằng số nên độ phức tạp của thuật toán EFI là hằng số.

Thực nghiệm trích xuất véc-tơ đặc trưng hình ảnh được tiến hành trên hai nhóm giá trị đó là véc-tơ gồm 189 chiều (SCH) và 225 chiều (SCH36), minh họa trên Hình 5 và Hình 6.



Hình 5. Trích xuất véc-tơ đặc trưng SCH



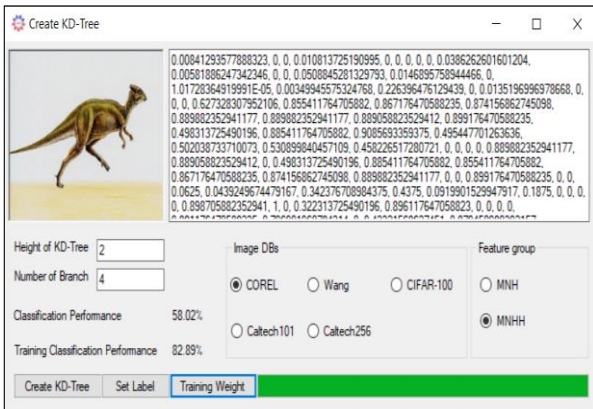
Hình 6. Trích xuất véc-tơ đặc trưng SCH36

4.2 Môi trường thực nghiệm

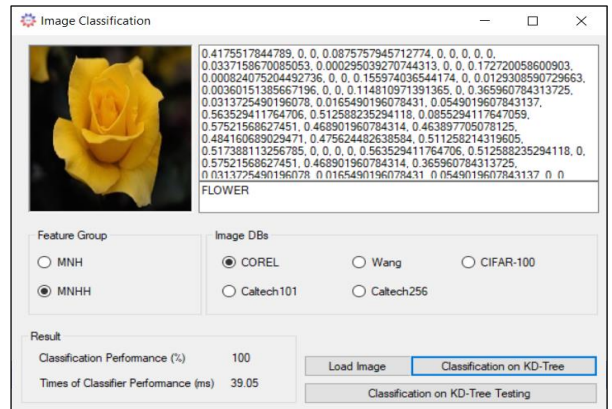
Để chứng minh tính đúng đắn của mô hình đề xuất, thực nghiệm được xây dựng trên nền tảng dotNET Framework 4.8, ngôn ngữ lập trình C#. Cấu hình máy tính của thực nghiệm pha kiểm thử: Intel(R) Core™ i7- 5200U, CPU 2,5 GHz, RAM 16 GB và hệ điều hành Windows 10 Professional. Cấu hình máy server dùng để huấn luyện mô hình phân lớp ảnh số dựa trên cấu trúc KD-Tree: CPU Xeon(R) Gold 6258R CPU 2,7 GHz, Ổ cứng SSD 1024 GB, RAM 16 GB, hệ điều hành Server Datacenter 2019.

4.3 Xây dựng thực nghiệm phân lớp ảnh trên cấu trúc KD-Tree

Thực nghiệm xây dựng cấu trúc KD-Tree theo phương pháp phân lớp được minh họa trên Hình 7. Trong thực nghiệm này, mỗi bộ ảnh có cấu trúc KD-Tree với chiều cao và số nhánh khác nhau để phù hợp với số phân lớp trên từng bộ ảnh. Hình 7 minh họa thực nghiệm xây dựng KD-Tree (Create KD-Tree) trên bộ ảnh COREL với 10 phân lớp nên chiều cao bằng 2 và số nhánh tối đa tại một node trên cây là 4. Với mỗi bộ ảnh, thực nghiệm chọn nhóm đặc trưng SCH hoặc SCH36. Sau khi xây dựng cấu trúc KD-Tree, thực hiện gán nhãn (Set Label) cho các nút lá trên KD-Tree để tính hiệu suất phân lớp ứng với bộ véc-tơ phân lớp xây dựng KD-Tree; huấn luyện tập véc-tơ phân lớp hình ảnh (Training Weight). Kết quả kiểm thử mô hình phân lớp ảnh số được minh họa trên Hình 8 (CL-KDT).



Hình 7. Xây dựng và huấn luyện cấu trúc KD-Tree



Hình 8. Thực nghiệm phân lớp ảnh bằng KD-Tree

4.4 Kết quả thực nghiệm và đánh giá hiệu suất mô hình phân lớp ảnh

Mô hình phân lớp ảnh đã đề xuất có hiệu suất phân lớp cho bộ dữ liệu kiểm thử là 30% trên các tập ảnh COREL, Wang, CIFAR-100, Caltech101 và Caltech256. Kết quả phân lớp được trình bày trong Bảng 2. Bộ véc-tơ đặc trưng 189 chiều và kết quả phân lớp các bộ ảnh thực nghiệm trên bộ véc-tơ đặc trưng 225 chiều. So sánh kết quả phân lớp với các kỹ thuật khác trên cùng bộ ảnh được trình bày trong Bảng 3.

Kết quả phân lớp hình ảnh trong Bảng 2 cho thấy với mỗi bộ đặc trưng được trích xuất khác nhau thì hiệu suất phân lớp trên các bộ ảnh cũng khác nhau. Điều này cho thấy loại đặc trưng và số lượng đặc trưng của hình ảnh có ảnh hưởng đến bài toán phân lớp ảnh. Sự khác nhau này phụ thuộc vào tính chất từng bộ ảnh cũng như phụ thuộc vào cấu trúc phân lớp KD-Tree khi xây dựng.

Để đạt được hiệu suất phân lớp trên các bộ ảnh thực nghiệm, một số chi phí về thời gian huấn luyện và số lần huấn luyện trên máy server có cấu hình CPU Xeon(R) Gold 6258R CPU 2,7 Ghz, ổ cứng SSD 1024 GB, RAM 16 GB, hệ điều hành Server Datacenter 2019 được trình bày trong Bảng 3.

Bảng 2. Hiệu suất phân lớp ảnh dựa trên cấu trúc KD-Tree với các bộ véc-tơ

Bộ dữ liệu	Số ảnh kiểm thử	Số phân lớp	Hiệu suất phân lớp với đặc trưng SCH	Hiệu suất phân lớp với đặc trưng SCH36
COREL	300	10	0,8232	0,8389
Wang	3.240	80	0,8089	0,8101
CIFAR-100	18.000	100	0,7959	0,7989
Caltech101	2.644	102	0,8109	0,8128
Caltech256	9.183	257	0,7952	0,8081

Bảng 3. Chi phí huấn luyện KD-Tree trên các bộ ảnh thực nghiệm với bộ đặc trưng SCH36

Bộ dữ liệu	Số Epoch	Tổng thời gian huấn luyện	Số lần huấn luyện
COREL	10	9:45:16	102
Wang	20	32:28:39	171
CIFAR-100	60	58:19:09	239
Caltech101	12	71:54:05	209
Caltech256	15	98:59:45	321

Để so sánh hiệu quả của phương pháp phân lớp hình ảnh tiếp cận KD-Tree với các phương pháp khác thực nghiệm trên cùng bộ ảnh, chúng tôi so sánh kết quả với các công trình khác (Bảng 4).

Bảng 4 so sánh kết quả phân lớp trên mô hình đề xuất với các công trình khác trên cùng bộ dữ liệu thực nghiệm với các phương pháp khác nhau. Trong bài báo này, kết quả hiệu suất phân lớp được thực hiện so sánh nhưng không so sánh chi phí thời gian huấn luyện mô hình phân lớp vì cấu hình máy tính không đồng nhất. Đồng thời, chi phí huấn luyện mô hình phân lớp KD-Tree chỉ thực hiện một lần và sau đó được thực thi nhiều lần. Kết quả so sánh với các công trình phân lớp ảnh trên cùng bộ dữ liệu cho thấy, phương pháp phân lớp ảnh CL-KDT bằng cấu trúc KD-Tree là hoàn toàn khả thi và hiệu quả. Có thể áp dụng phương pháp này cho bài toán phân loại đối tượng qua hình ảnh, nhận dạng mẫu và áp dụng cho bài toán tìm kiếm ảnh tương tự.

Bảng 4. So sánh kết quả phân lớp trên các bộ ảnh với các công trình khác

Công trình	Bộ dữ liệu	Phương pháp	Hiệu suất phân lớp
Páváloi et al. [13]	COREL	kNN ($k = 4$)	0,7990
Aslam et al. [23]	COREL	BoVW, Mid-level features	0,8126
CL-KDT	COREL	KD-Tree- SCH36	0,8389
Nhi et al. [5]	Wang	C-Tree and kNN	0,6894
Kumar et al. [15]	Wang	SVM-NPP-SURF	0,7900
CL-KDT	Wang	KD-Tree- SCH36	0,8101
Kumar et al. [15]	CIFAR-100	CNN	0,5802
Rashid et al. [16]	CIFAR-100	ESD	0,6800
CL-KDT	CIFAR-100	KD-Tree- SCH36	0,7989
Dang et al. [17]	Caltech101	k-NN	0,7800
Kabbai et al. [18]	Caltech101	SURF-Wavelet Transform	0,7990
CL-KDT	Caltech101	KD-Tree- SCH36	0,8128
Zheng et al. [19]	Caltech256	Enhanced visual tree	0,6406
Li [20]	Caltech256	k-NN	44,42
CL-KDT	Caltech256	KD-Tree-SCH36	0,8081

5 Kết luận và hướng phát triển

Trong bài báo này, chúng tôi đã tiến hành một phương pháp phân lớp hình ảnh bằng cấu trúc KD-Tree đa nhánh cân bằng. Thực nghiệm được xây dựng trên các bộ ảnh COREL, Wang, CIFAR-100, Caltech101 và Caltech256 để chứng minh tính khả thi của phương pháp được chúng tôi đề xuất. Kết quả thực nghiệm được đánh giá dựa trên độ chính xác phân lớp trên tập ảnh Testing độc lập với bộ ảnh Training; đồng so sánh với các công trình đã công bố về kết quả phân lớp ảnh bằng các kỹ thuật khác trên cùng bộ dữ liệu thực nghiệm. Độ chính xác phân lớp trung bình tương ứng với từng bộ ảnh COREL, Wang, CIFAR-100, Caltech101 và Caltech256 nhóm đặc trưng SCH36 lần lượt là 0,8389, 0,8101, 0,7989, 0,8109 và 0,8081. Kết quả này cho thấy phương pháp đề xuất là hiệu quả và có thể áp dụng được cho các hệ phân loại ảnh thuộc các lĩnh vực khác nhau. Hướng phát triển tiếp theo là chúng tôi sẽ xây dựng một mô hình phân lớp ảnh dựa trên cấu trúc KD-Tree kết hợp với các kỹ thuật học máy k-NN, SVM và Random Forest nhằm tăng hiệu suất phân lớp hình ảnh.

Lời cảm ơn

Chúng tôi trân trọng cảm ơn Khoa Công nghệ thông tin, Trường Đại học Khoa học, Đại học Huế, nhóm nghiên cứu SBIR-HCM, Trường Đại học Sư phạm Tp. HCM và Trường Đại học Công nghiệp Thực phẩm Thành phố Hồ Chí Minh đã hỗ trợ về chuyên môn và cơ sở vật chất để nhóm tác giả hoàn thành nghiên cứu này.

Tài liệu tham khảo

1. Patrizio, Data center explorer, Network World. <https://www.networkworld.com/article/3325397/idc-expect-175-zettabytes-of-data-worldwide-by-2025.html>, 1/3/2022
2. Sawant Shrutika S. and M. Prabukumar (2017), Semi-supervised techniques based hyper-spectral image classification: a survey, *Innovations in Power and Advanced Computing Technologies (i-PACT) IEEE*.
3. Zhang, Jiayu and Hongbo Shi (2019), Kd-tree based efficient ensemble classification algorithm for imbalanced learning, *international conference on machine learning, big data and business intelligence (MLDBI) IEEE*.
4. Bentley, J. L (1975), Multidimensional binary search trees used for associative searching, *Communications of the ACM*, 18(9), 509-517.
5. Nhi, Nguyen Thi Uyen, and Thanh Manh Le, (2021), Semantic-Based Image Retrieval Using Balanced Clustering Tree, *World Conference on Information Systems and Technologies. Springer, Cham*.
6. Nguyễn Thị Định, Thế Thành Văn, Mạnh Thanh Lê, (2021), Phân lớp ảnh bằng cây KD-Tree cho bài toán tìm kiếm ảnh tương tự, *Các công trình nghiên cứu, phát triển và ứng dụng Công nghệ Thông tin và Truyền thông*, 40-52.
7. Nguyễn Thị Định, Thế Thành Văn, Mạnh Thanh Lê (2021), Một phương pháp phân lớp trên cấu trúc KD-Tree cho bài toán tìm kiếm ảnh theo ngữ nghĩa, *Kỷ yếu Hội thảo Quốc gia về Nghiên cứu cơ bản và ứng dụng CNTT (FAIR21), ĐH Công nghiệp Thực phẩm TP. HCM, Nhà xuất bản Khoa học Tự nhiên và Công nghệ, ISBN 978-604-9988-60-8. DOI: 10.15625/vap.2021.0075*
8. Corel 1k Database: <http://wang.ist.psu.edu/docs/related/>, 21/12/2021
9. Wang Database: <http://wang.ist.psu.edu/docs/home.shtml>, 21/12/2021
10. CIFAR-100 Database: <https://www.cs.toronto.edu/~kriz/cifar.html>, 21/12/2021
11. Caltech101: <http://www.vision.caltech.edu/Images Dataset/Caltech101>, 21/12/2021
12. Caltech256: <https://www.kaggle.com/jessicali9530/caltech256>, 21/12/2021
13. Păvăloi, Ioan and Cristina Diana Nită, (2017), Experiments on image classification and retrieval using statistics on pixels position, *International Symposium on Signals, Circuits and Systems (ISSCS) IEEE*.
14. Ouni (2020), A machine learning approach for image retrieval tasks, *35th International Conference on Image and Vision Computing New Zealand (IVCNZ). IEEE*.
15. Kumar et al. Image Classification Using Convolution Neural Network and Comparative Analysis.
16. Rashid et al. (2020), A sustainable deep learning framework for object recognition using multi-layers deep features fusion and selection, *Sustainability*, 5037.

17. Dang et al. (2018), Image classification based on quantum K-Nearest-Neighbor algorithm, *Quantum Information Processing*, 17(9), 1-18.
18. Kabbai, Leila, Mehrez Abdellaoui and Ali Douik (2019), Image classification by combining local and global features, *The Visual Computer*, 35(5), 679-693.
19. Zheng et al. (2017), Hierarchical learning of multi-task sparse metrics for large-scale image classification, *Pattern Recognition*, 67, 97-109.
20. Li et al. (2018), Improving image classification accuracy with ELM and CSIFT, *Computing in Science & Engineering*, 21(5), 26-34.
21. Zhang et al. (2016), Fast face sketch synthesis via kd-tree search, *European Conference on Computer Vision*. Springer, Cham.
22. Khotimah, Wijayanti Nurul, et al. (2015), Tuna fish classification using decision tree algorithm and image processing method, *2015 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*, IEEE.
23. Aslam, M. A., Salik, M. N., Chughtai, F., Ali, N., Dar, S. H. & Khalil, T. (2019, December), Image classification based on mid-level feature fusion, *In 2019 15th International Conference on Emerging Technologies (ICET) IEEE* (1-6).
24. Gautam et al. (2016), Automatic classification of leukocytes using morphological features and naïve Bayes classifier, *2016 IEEE region 10 conference (TENCON)*, IEEE.